EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	105	714/6.ccls. and @pd>="20070520"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/09/14 14:30
L2	18	714/8.ccls. and @pd>="20070520"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/09/14 13:32
L4	6	714/715.ccls. and @pd>="20070520"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/09/14 13:33
L5	3	714/797.ccls. and @pd>="20070520"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/09/14 13:33
L6	80	365/200.ccls. and @pd>="20070520"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/09/14 13:45
L7	351	(memory with test) and (cpu or processor or "processing unit") and ((microprocessor or microcontroller) with embedded with memory)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/09/14 14:50
L8	1007	714/30.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/09/14 14:47
L9	451	714/30.ccls. and (memory with test)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/09/14 14:38
L10	111	714/30.ccls. and ("memory test")	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/09/14 14:39

EAST Search History

L11	820	714/42.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/09/14 14:38
L12	110	714/42.ccls. and ("memory test")	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/09/14 14:39
S1	1	10/674044	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/04 13:32
S2	903	714/5.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/04 13:33
S3	54	S2 AND (self-test$3 OR (self ADJ test$3))	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/25 13:48
S4	37	S3 AND RAM	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/04 13:34
S5	1755	714/6.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/04 14:05
S6	228	S5 AND threshold	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/04 14:05

EAST Search History

S7	7	S6 AND 714/8.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/04 14:05
S8	19	("5208813" \| "5526482" \| "5588110" \| "5724501" \| "5905858" \| "6023748" \| "6038680" \| "6052798" \| "6073218").PN. OR ("6505305"). URPN.	US-PGPUB; USPAT; USOCR	OR	ON	2006/05/04 14:09
S9	2	"5862314".pn.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/04 14:25
S10	20896	detect$3 WITH (fault OR error) WITH memory	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/04 14:26
S11	9	S5 AND (read ADJ test)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/04 14:30
S12	719	ECC WITH (mirror$3 OR redundant$2)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/04 14:31
S13	2	ECC WITH (mirror$3 OR redundant$2) WITH (benefit$1 OR improvement$1)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/04 14:37
S14	674	714/42.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/04 14:38

EAST Search History

S15	34	S14 AND (read ADJ test)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/04 14:38
S16	2043	(memory WITH test$3) AND (read WITH (write OR written) WITH pattern)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/04 15:23
S17	1071	(memory WITH test$3) AND (read WITH (write OR written) WITH pattern) AND (correct$4)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/04 15:23
S18	204	(memory WITH test$3) AND (read WITH (write OR written) WITH pattern SAME correct$4)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/04 15:24
S19	3	S18 AND S2	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/04 15:27
S20	415	714/8.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/04 15:27
S21	269	S20 AND correct$3	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/04 15:29
S22	3958	365/200.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/04 15:30

EAST Search History

S23	9	S21 AND (test ADJ pattern)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/04 15:52
S24	333	(write OR written OR writing) NEAR (test ADJ pattern) WITH read	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/04 15:52
S25	119	S24 AND (fix$3 OR corect$3)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/04 15:46
S26	117	S25 AND (memory OR disk OR RAM)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/04 15:35
S27	0	S24 AND (corect$3)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/04 15:46
S28	4474	detect$3 WITH correct$3 WITH (bit NEAR (error OR fault))	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/04 15:47
S29	4	S28 WITH (test ADJ pattern)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/04 15:48
S30	415	714/8.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/04 15:49

EAST Search History

S31	75	S30 AND (correct$3 OR recover$3) ADJ (error OR fault)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/04 15:49
S32	162	S30 AND (correct$3 OR recover$3) NEAR (error OR fault)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/04 15:49
S33	60	S24 AND recover$3	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/04 15:51
S34	12	S24 AND (recover$3.AB.)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/04 15:51
S35	93	((write OR written OR writing) NEAR (test ADJ pattern) WITH read).AB.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/04 15:52
S36	2	07/890549	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/04 16:09
S37	215	714/715.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/04 16:25
S38	0	S37 WITH (correct$3)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/04 16:25

EAST Search History

S39	124	S37 AND (correct$3)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/04 16:27
S40	29	S37 AND (correct$3) AND RAM	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/04 16:27
S41	0	voting WITH ((replace OR replacing OR replacement) NEAR bit)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/04 16:50
S42	0	vot$3 WITH ((replace OR replacing OR replacement) NEAR bit)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/04 16:50
S43	0	vot$3 WITH ((replace OR replacing OR replacement) NEAR2 bit)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/04 16:51
S44	366	714/797.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/04 16:52
S45	20	S44 AND ECC	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/04 16:53
S46	76	mirror$3 SAME (vote OR voting)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/04 16:53

EAST Search History

S47	44	S46 AND (memory OR storage)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/04 16:54
S48	35	S46 AND (memory OR storage) NOT election	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/05 09:56
S49	86	self ADJ testing ADJ (memory OR storage)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/05 08:06
S50	31	S49 AND RAM	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/05 08:06
S51	1	mirroring WITH voting	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/05 09:03
S52	97	redundant WITH data WITH voting	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/05 09:04
S53	39	redundant ADJ data WITH voting	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/05 09:05
S54	24	S53 AND (disk Or storage)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/05 09:06

EAST Search History

S55	1755	714/6.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/05 09:07
S56	20	S55 AND voting	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/05 09:57
S57	0	S49 AND (vote$1 OR voting)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/05 09:56
S58	903	714/5.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/05 09:56
S59	14	S58 AND voting	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/05 10:01
S60	555	(mirror$3 OR redundan$3) WITH voting	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/05 10:02
S61	22	((mirror$3 OR redundan$3) WITH voting) SAME (disk OR (storage ADJ (device OR drive)))	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/05 10:03
S62	1	"3896416".PN.	USPAT; USOCR	OR	ON	2006/05/05 10:42
S63	19	("3896416" \| "4464754" \| "4464755" \| "4670880" \| "4873685" \| "5056095" \| "5067132" \| "5267250").PN. OR ("5907671"). URPN.	US-PGPUB; USPAT; USOCR	OR	ON	2006/05/05 10:49

EAST Search History

S64	0	((memory OR storage) ADJ test$3) WITH (vote OR voting)	US-PGPUB; USPAT; USOCR	OR	ON	2006/05/05 10:51
S65	635	((memory OR storage) ADJ test$3). AB.	US-PGPUB; USPAT; USOCR	OR	ON	2006/05/05 10:51
S66	1257	((memory OR storage) NEAR test$3).AB.	US-PGPUB; USPAT; USOCR	OR	ON	2006/05/05 10:51
S67	10	S66 AND (vote OR voting)	US-PGPUB; USPAT; USOCR	OR	ON	2006/05/05 10:52
S68	1	"6879530".pn.	US-PGPUB; USPAT; USOCR	OR	ON	2006/05/05 10:52
S69	14150	((self-test$3 OR (self ADJ test$3) NEAR RAM))	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/25 13:49
S70	3153	((self-test$3 OR (self ADJ test$3) NEAR RAM)).AB.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/25 15:04
S71	1	10/944990	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/25 14:35
S72	0	("2005/0283566").URPN.	USPAT	OR	ON	2006/10/25 14:35
S73	995	714/5.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/25 15:04
S74	5	S70 AND S73	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/25 15:04

EAST Search History

S75	17	("20020046377" \| "20020174394" \| "20030016578" \| "4333142" \| "5222066" \| "5388104" \| "5475815" \| "5535164" \| "5617531" \| "5835502" \| "5974579" \| "5995731" \| "6085346" \| "6163490" \| "6209074" \| "6360342" \| "6446184").PN. OR ("6993692"). URPN.	US-PGPUB; USPAT; USOCR	OR	ON	2006/10/25 15:21
S76	44	("4459685" \| "4523313" \| "4598388" \| "4601019" \| "4686737" \| "5031151" \| "5257229" \| "5268866" \| "5270976" \| "5287310" \| "5293383" \| "5355340" \| "5396124" \| "5422850" \| "5471426" \| "5471482" \| "5502674" \| "5506959" \| "5511028" \| "5544106" \| "5572470" \| "5572471" \| "5574732" \| "5583463" \| "5748543" \| "5764577" \| "5835504" \| "5940335" \| "5946246" \| "5963489" \| "5978297" \| "5978931" \| "6005813" \| "6064617" \| "6125067" \| "6182257" \| "6199177" \| "6216241" \| "6243301" \| "6300769" \| "6330696" \| "6343366" \| "6351837").PN. OR ("6879530"). URPN.	US-PGPUB; USPAT; USOCR	OR	ON	2006/10/27 09:00
S77	3	S76 AND remap$4	US-PGPUB; USPAT; USOCR	OR	ON	2006/10/27 09:31
S78	961	714/5.ccls.	US-PGPUB; USPAT; USOCR	OR	ON	2006/10/27 09:31
S79	48	S78 AND remap$4	US-PGPUB; USPAT; USOCR	OR	ON	2006/10/27 09:35
S80	1	08/741643	US-PGPUB; USPAT; USOCR	OR	ON	2006/10/27 09:35
S81	5	08/741603	US-PGPUB; USPAT; USOCR	OR	ON	2006/10/27 09:35
S82	0	2004/0161257	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/30 13:54

EAST Search History

S83	2	"20040161257"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/30 15:40
S84	44	("4459685" \| "4523313" \| "4598388" \| "4601019" \| "4686737" \| "5031151" \| "5257229" \| "5268866" \| "5270976" \| "5287310" \| "5293383" \| "5355340" \| "5396124" \| "5422850" \| "5471426" \| "5471482" \| "5502674" \| "5506959" \| "5511028" \| "5544106" \| "5572470" \| "5572471" \| "5574732" \| "5583463" \| "5748543" \| "5764577" \| "5835504" \| "5940335" \| "5946246" \| "5963489" \| "5978297" \| "5978931" \| "6005813" \| "6064617" \| "6125067" \| "6182257" \| "6199177" \| "6216241" \| "6243301" \| "6300769" \| "6330696" \| "6343366" \| "6351837").PN. OR ("6879530"). URPN.	US-PGPUB; USPAT; USOCR	OR	ON	2006/11/01 17:18
S88	1037	714/5.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/01/26 13:56
S89	118	S88 AND (divid$3 WITH memor$3)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/01/26 14:08
S90	0	S88 AND (faster WITH test$3).AB.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/01/26 14:15
S91	751	714/42.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/01/26 14:17

EAST Search History

S92	8359	(714/5,6,8,42,797.ccls.) OR 365/200.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/01/26 14:18
S93	58	S92 AND faster WITH test$3	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/01/26 14:19
S94	273	S92 AND divid$3 WITH access	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/01/26 15:24
S95	1	S92 AND simultaneuos WITH access	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/01/26 15:24
S96	1	S92 AND simultaneuos$2 WITH access	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/01/26 15:25
S97	201	S92 AND concurrent$2 WITH access	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/01/26 15:24
S98	241	S92 AND concurrent$2 WITH access$3	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/01/26 15:24
S99	1	S92 AND simultaneuos$2 WITH access$3	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/01/26 16:06
S100	5	(("5751729") or ("5570374") or ("4939694") or ("6041426") or ("5036516")).PN.	USPAT; USOCR	OR	OFF	2007/01/26 15:40

EAST Search History

S101	61	S92 AND (concurrent$2 OR simultaneuos$2) WITH test$3	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/01/26 16:07
S102	177	714/5.ccls. and @pd>="20060505"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/20 13:06
S103	28	714/5.ccls. and @pd>="20060505" and (memory with test)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/20 13:06
S104	450	714/6.ccls. and @pd>="20060505"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/20 13:14
S105	23	714/6.ccls. and @pd>="20060505" and (memory with test)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/09/14 10:04
S106	65	714/8.ccls. and @pd>="20060505"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/20 13:15
S107	10	714/8.ccls. and @pd>="20060505" and (memory with test)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/20 13:16
S108	29	714/715.ccls. and @pd>="20060505"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/20 13:16
S109	10	714/715.ccls. and @pd>="20060505" and (memory with test)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/20 13:16

EAST Search History

S110	9	714/797.ccls. and @pd>="20060505"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/20 13:16
S111	12	("6968397" "6738878" "6564319" "7073064" "7069389" "5535328").pn.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/20 13:20
S112	64506	(memory with test)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/20 13:44
S113	323	(memory with test) and (cpu or processor or "processing unit") and ((microprocessor or microcontroller) with embedded with memory)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/09/14 14:20
S114	37	714/5.ccls. and @pd>="20070520"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/09/14 12:39

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L13	214	714/30.ccls.	US-PGPUB	OR	ON	2007/09/14 14:47
L14	143	(memory with test) and (cpu or processor or "processing unit") and ((microprocessor or microcontroller) with embedded with memory)	US-PGPUB	OR	ON	2007/09/14 14:50

Scholarly articles for ("memory test") and (cpu or processor or "processing unit") and ((microprocessor or microcontroller) and embedded and memory)

System level functional testing through one or more I/O ... - Iqbal - Cited by 24

Patents in Class 714/29

A **microcontroller** including a system bus; a **microprocessor** coupled to the ... The internal trace **memory** is **embedded** in an ICE **CPU** to trace **CPU** internal ...
www.freepatentsonline.com/CCL-714-29.html - 62k - Cached - Similar pages

Initialization of flash storage via an **embedded** controller ...

6009496 describes a **microcontroller** architecture with **embedded** flash **memory**, in which an on-board **microprocessor** executes a reprogramming control routine, ...
www.freepatentsonline.com/20070088940.html - 67k - Cached - Similar pages
[More results from www.freepatentsonline.com]

Generating notification that a new **memory** module has been added to ...

A **memory test** of the new **memory** module may also be indicated. ... The control unit or **processor** includes a **microprocessor**, a **microcontroller**, a **processor** ...
www.patentstorm.us/patents/6996648-description.html - 30k - Cached - Similar pages

Method for operating application software in a safety critical ...

The computing system includes the central **processing unit** 15 which is preferably a **microprocessor or microcontroller**, such as an INTEL 486 or Pentium ...
www.patentstorm.us/patents/5572671-description.html - 43k - Cached - Similar pages

Error detection/correction and fault detection/recovery patents 200612

A **microcontroller** comprises a **microprocessor** and a **memory** device ... The **microcontroller** includes a central **processor** unit (**CPU**) and a Flash program **memory** ...
www.freshpatents.com/Error-detection-correction-and-fault-detection-recovery-dt200612ntc714.php - 97k - Cached - Similar pages

Error detection/correction and fault detection/recovery patents 200603

20060059386 - Testing **embedded memory** in an integrated circuit: An integrated When the central **processing unit** of the microcomputer or **microcontroller** ...
www.freshpatents.com/Error-detection-correction-and-fault-detection-recovery-dt200603ntc714.php - 97k - Cached - Similar pages
[More results from www.freshpatents.com]

[PDF] **Microprocessor** testability - Industrial Electronics, IEEE ...

a **microprocessor** was. Motorola with the MC6802. [19]. This part runs its own on-. chip **memory test**. The experimental data given in Table I ...
ieeexplore.ieee.org/iel1/41/723/00019064.pdf?arnumber=19064 - Similar pages

Sitemap

AMD Sempron **Processor** 2100+ and Geode LX 800 for **Embedded** System · AMD LIVE! Freescale Released MPC5566 **Microcontroller** with 3MB of Flash **Memory** ...
embedded-system.net/sitemap.html - 219k - Cached - Similar pages

EP1388788 St european software patent - Built-in self test circuit ...
Clearly, the **memory** 103 might be one of several subsystems **embedded** in the IC 101: ... **microprocessor** 105, including a central **processing unit** (**CPU**) 107, ...
gauss.ffii.org/PatentView/EP1388788 - 84k - Cached - Similar pages

Participants
ThumbPods are a class of systems which contain a **microcontroller**, a CMOS fingerprint ... **Embedded memory test** design has become an essential part of the ...
users.ece.gatech.edu/limsk/ubooth/detail03.html - 91k - Cached - Similar pages

1 2 3 4 5 6 7 8 9 10 **Next**

Try Google Desktop: search your computer as easily as you search the web.

("memory test") and (cpu or process Search

Search within results | Language Tools | Search Tips | Dissatisfied? Help us improve

©2007 Google - Google Home - Advertising Programs - Business Solutions - About Google

PORTAL
USPTO

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: The ACM Digital Library The Guide

("memory test") and (cpu or processor or "processing unit") ar SEARCH

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Terms used: **memory test** and **cpu** or **processor** or **processing unit** and **microprocessor** or **microcontroller** and **embedded** and **memory**

Found **35,646** of **210,707**

Sort results by relevance

Display results expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown

Relevance scale

1 A survey of processors with explicit multithreading

Theo Ungerer, Borut Robič, Jurij Šilc

March 2003 **ACM Computing Surveys (CSUR)**, Volume 35 Issue 1

Publisher: ACM Press

Full text available: pdf(920.16 KB) Additional Information: full citation, abstract, references, citings, index terms

Hardware multithreading is becoming a generally applied technique in the next generation of microprocessors. Several multithreaded processors are announced by industry or already into production in the areas of high-performance microprocessors, media, and network processors.A multithreaded processor is able to pursue two or more threads of control in parallel within the processor pipeline. The contexts of two or more threads of control are often stored in separate on-chip register sets. Unused i ...

Keywords: Blocked multithreading, interleaved multithreading, simultaneous multithreading

2 System-level power optimization: techniques and tools

Luca Benini, Giovanni de Micheli

April 2000 **ACM Transactions on Design Automation of Electronic Systems (TODAES)**, Volume 5 Issue 2

Publisher: ACM Press

Full text available: pdf(385.22 KB) Additional Information: full citation, abstract, references, citings, index terms

This tutorial surveys design methods for energy-efficient system-level design. We consider electronic sytems consisting of a hardware platform and software layers. We consider the three major constituents of hardware that consume energy, namely computation, communication, and storage units, and we review methods of reducing their energy consumption. We also study models for analyzing the energy cost of software, and methods for energy-efficient software design and compilation. This survery ...

3 Network processors: a perspective on market requirements, processor architectures and embedded S/W tools

P. Paulin, F. Karim, P. Bromley

March 2001 **Proceedings of the conference on Design, automation and test in Europe DATE '01**

Publisher: IEEE Press
Full text available: pdf(269.19 KB) Additional Information: full citation, references, citings, index terms

4 Processor-programmable memory BIST for bus-connected embedded memories
Ching-Hong Tsai, Cheng-Wen Wu
January 2001 **Proceedings of the 2001 conference on Asia South Pacific design automation ASP-DAC '01**
Publisher: ACM Press
Full text available: pdf(73.62 KB) Additional Information: full citation, abstract, references, index terms

We present a processor-programmable built-in self-test (BIST) scheme suitable for embedded memory testing in the system-on-a-chip (SOC) environment. The proposed BIST circuit can be programmed via an on-chip microprocessor. Upon receiving the commands from the microprocessor, the BIST circuit generates pre-defined test patterns and compares the memory outputs with the expected outputs. Most popular memory test algorithms can be realized by properly programming the BIST circuit using the pro ...

5 A performance analysis of PIM, stream processing, and tiled processing on memory-intensive signal processing kernels
Jinwoo Suh, Eun-Gyu Kim, Stephen P. Crago, Lakshmi Srinivasan, Matthew C. French
May 2003 **ACM SIGARCH Computer Architecture News , Proceedings of the 30th annual international symposium on Computer architecture ISCA '03**, Volume 31 Issue 2
Publisher: ACM Press
Full text available: pdf(239.50 KB) Additional Information: full citation, abstract, references, citings

Trends in microprocessors of increasing die size and clock speed and decreasing feature sizes have fueled rapidly increasing performance. However, the limited improvements in DRAM latency and bandwidth and diminishing returns of increasing superscalar ILP and cache sizes have led to the proposal of new microprocessor architectures that implement processor-in- memory, stream processing, and tiled processing. Each architecture is typically evaluated separately and compared to a baseline architectu ...

6 Emerging areas: Fault-tolerant platforms for automotive safety-critical applications
M. Baleani, A. Ferrari, L. Mangeruca, A. Sangiovanni-Vincentelli, Maurizio Peri, Saverio Pezzini
October 2003 **Proceedings of the 2003 international conference on Compilers, architecture and synthesis for embedded systems CASES '03**
Publisher: ACM Press
Full text available: pdf(736.40 KB) Additional Information: full citation, abstract, references, citings, index terms

Fault-tolerant electronic sub-systems are becoming a standard requirement in the automotive industrial sector as electronics becomes pervasive in present cars. We address the issue of fault tolerant chip architectures for automotive applications. We begin by reviewing fault-tolerant architectures commonly used in other industrial domains where fault-tolerant electronics has been a must for a number of years, e.g., the aircraft manufacturing industrial sector. We then proceed to investigate how t ...

Keywords: VLSI, automotive, fault-tolerant, multi-processor, safety critical, system-on-a-chip

7 Meeting the Embedded Design Needs of Automotive Applications
Wayne Lyons

March 2005 **Proceedings of the conference on Design, Automation and Test in Europe - Volume 3 DATE '05**
Publisher: IEEE Computer Society
Full text available: pdf(205.16 KB) Additional Information: full citation, abstract, citings, index terms

The importance of embedded systems in driving innovation in automotive applications continues to grow. Understanding the specific needs of developers targeting this market is also helping to drive innovation in RISC core design. This paper describes how a RISC instruction set architecture has evolved to better meet those needs, and the key implementation features in two very different RISC cores are used to demonstrate the challenges of designing for real-time automotive systems.

8 A P1500-Compatible Programmable BIST Aapproach for the Test of Embedded Flash Memories
P. Bernardi, M. Rebaudengo, M. Sonza Reorda, M. Violante
March 2003 **Proceedings of the conference on Design, Automation and Test in Europe - Volume 1 DATE '03**
Publisher: IEEE Computer Society
Full text available: pdf(154.45 KB) Publisher Site Additional Information: full citation, abstract, index terms

In this paper we present a microprocessor-based approach suitable for embedded flash memory testing in a System-on-a-chip (SOC) environment. The main novelty of the approach is the high flexibility, which guarantees easy exploitation of the same architecture to different memory cores. The proposed approach is compatible with the P1500 standard. A case study has been developed and demonstrates the advantages of the proposed core test strategy in terms of area overhead and test application time.

9 Spatial computation
Mihai Budiu, Girish Venkataramani, Tiberiu Chelcea, Seth Copen Goldstein
October 2004 **ACM SIGARCH Computer Architecture News , ACM SIGPLAN Notices , ACM SIGOPS Operating Systems Review , Proceedings of the 11th international conference on Architectural support for programming languages and operating systems ASPLOS-XI**, Volume 32 , 39 , 38 Issue 5 , 11 , 5
Publisher: ACM Press
Full text available: pdf(573.00 KB) Additional Information: full citation, abstract, references, citings, index terms, review

This paper describes a computer architecture, *Spatial Computation* (SC), which is based on the translation of high-level language programs directly into hardware structures. SC program implementations are completely distributed, with no centralized control. SC circuits are optimized for *wires* at the expense of computation units.In this paper we investigate a particular implementation of SC: ASH (Application-Specific Hardware). Under the assumption that computation is cheaper than co ...

Keywords: application-specific hardware, dataflow machine, low-power, spatial computation

10 Security in embedded systems: Design challenges
Srivaths Ravi, Anand Raghunathan, Paul Kocher, Sunil Hattangady
August 2004 **ACM Transactions on Embedded Computing Systems (TECS)**, Volume 3 Issue 3
Publisher: ACM Press
Full text available: pdf(3.67 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Many modern electronic systems---including personal computers, PDAs, cell phones,

network routers, smart cards, and networked sensors to name a few---need to access, store, manipulate, or communicate sensitive information, making security a serious concern in their design. Embedded systems, which account for a wide range of products from the electronics, semiconductor, telecommunications, and networking industries, face some of the most demanding security concerns---on the one hand, they are oft ...

Keywords: Embedded systems, architecture, authentication, battery life, cryptographic algorithms, decryption, encryption, hardware design, processing requirements, security, security attacks, security protocols, tamper resistance

11 Test: A constraint-based solution for on-line testing of processors embedded in real-time applications

Marcelo Moraes, Érika Cota, Luigi Carro, Flávio Wagner, Marcelo Lubaszewski
September 2005 **Proceedings of the 18th annual symposium on Integrated circuits and system design SBCCI '05**
Publisher: ACM Press
Full text available: pdf(133.86 KB) Additional Information: full citation, abstract, references, index terms

Software-based self-test has been proposed as a low-cost strategy for on-line periodic testing of embedded processors. In this paper, we show that structural test programs composed only by regular deterministic self-test routines may be unfeasible in a real-time embedded platform. Hence, we propose a method to consciously select a set of test routines from different test approaches to compose a test program for an embedded processor. The proposed method not only ensures the periodical execution ...

Keywords: embedded processors, on-line testing, real-time systems, software-based self-test, test space exploration

12 On the hardware-software partitioning problem: System modeling and partitioning techniques

Marisa López-Vallejo, Juan Carlos López
July 2003 **ACM Transactions on Design Automation of Electronic Systems (TODAES)**, Volume 8 Issue 3
Publisher: ACM Press
Full text available: pdf(409.78 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper presents an in-depth study of several system partitioning procedures. It is based on the appropriate formulation of a general system model, being therefore independent of either the particular co-design problem or the specific partitioning procedure. The techniques under study are a knowledge-based system and three classical circuit partitioning algorithms (Simulated Annealing, Kernighan&Lin and Hierarchical Clustering). The former has been entirely proposed by the authors in previous ...

Keywords: Hardware-software co-design, clustering, cost functions, expert systems, fuzzy logic, general optimization procedures, hardware-software partitioning, system modeling

13 System-level power optimization: techniques and tools

Luca Benini, Giovanni De Micheli
August 1999 **Proceedings of the 1999 international symposium on Low power electronics and design ISLPED '99**
Publisher: ACM Press
Full text available: pdf(663.71 KB) Additional Information: full citation, references, citings, index terms

14 Hardware/Software Co-testing of Embedded Memories in Complex SOCs
Bai Hong Fang, Qiang Xu, Nicola Nicolici
November 2003 **Proceedings of the 2003 IEEE/ACM international conference on Computer-aided design ICCAD '03**
Publisher: IEEE Computer Society
Full text available: pdf(145.29 KB) Additional Information: full citation, abstract, index terms

A novel approach for testing embedded memories in complexsystems-on-a-chip (SOCs) is presented. The proposedsolution aims to balance the usage of the existing on-chipresources and dedicated design for test (DFT) hardwaresuch that the functional power constraints are not exceededduring test while trading-off the testing time againstDFT area and performance overhead. The suitability ofsoftware-centric and hardware-centric approaches for embeddedmemory testing is examined and to combine the advanta ...

15 Software thread integration for embedded system display applications
Alexander G. Dean
February 2006 **ACM Transactions on Embedded Computing Systems (TECS)**, Volume 5 Issue 1
Publisher: ACM Press
Full text available: pdf(1.40 MB) Additional Information: full citation, abstract, references, index terms

Embedded systems require control of many concurrent real-time activities, leading to system designs that feature a variety of hardware peripherals, with each providing a specific, dedicated service. These peripherals increase system size, cost, weight, and design time. Software thread integration (STI) provides low-cost thread concurrency on general-purpose processors by automatically interleaving multiple threads of control into one. This simplifies hardware to software migration (which elimina ...

Keywords: fine-grain concurrency, hardware to software migration, software thread integration

16 Developing for the Atmel AVR microcontroller on Linux
Pat Deegan
February 2005 **Linux Journal**, Volume 2005 Issue 130
Publisher: Specialized Systems Consultants, Inc.
Full text available: html(25.18 KB) Additional Information: full citation, abstract, index terms

Make your microcontroller projects go smoothly with familiar, powerful GNU tools.

17 Networked sensor platforms: XYZ: a motion-enabled, power aware sensor node platform for distributed sensor network applications
Dimitrios Lymberopoulos, Andreas Savvides
April 2005 **Proceedings of the 4th international symposium on Information processing in sensor networks IPSN '05**
Publisher: IEEE Press
Full text available: pdf(188.45 KB) Additional Information: full citation, abstract, references, citings

This paper describes the XYZ, a new open-source sensing platform specifically designed to support our experimental research in mobile sensor networks. The XYZ node is designed around the OKI ML67Q500x ARM THUMB Microprocessor and the IEEE 802.15.4 compliant CC2420 radio from Chipcon. Its new features include support for two different CPU sleep modes and a long-term ultra low power sleep mode for the entire node. This allows the XYZ and its peripheral boards to transition into deep sleep for exte ...

18 Missing the memory wall: the case for processor/memory integration

Ashley Saulsbury, Fong Pong, Andreas Nowatzyk

May 1996 **ACM SIGARCH Computer Architecture News , Proceedings of the 23rd annual international symposium on Computer architecture ISCA '96**, Volume 24 Issue 2

Publisher: ACM Press

Full text available: pdf(1.45 MB) Additional Information: full citation, abstract, references, citings, index terms

Current high performance computer systems use complex, large superscalar CPUs that interface to the main memory through a hierarchy of caches and interconnect systems. These CPU-centric designs invest a lot of power and chip area to bridge the widening gap between CPU and main memory speeds. Yet, many large applications do not operate well on these systems and are limited by the memory subsystem performance.This paper argues for an integrated system approach that uses less-powerful CPUs that are ...

19 Logical and physical design issues for smart card databases

Cristiana Bolchini, Fabio Salice, Fabio A. Schreiber, Letizia Tanca

July 2003 **ACM Transactions on Information Systems (TOIS)**, Volume 21 Issue 3

Publisher: ACM Press

Full text available: pdf(1.12 MB) Additional Information: full citation, abstract, references, citings, index terms

The design of very small databases for smart cards and for portable embedded systems is deeply constrained by the peculiar features of the physical medium. We propose a joint approach to the logical and physical database design phases and evaluate several data structures with respect to the performance, power consumption, and endurance parameters of read/program operations on the Flash-EEPROM storage medium.

Keywords: Design methodology, access methods, data structures, flash memory, personal information systems, smart card

20 Energy-aware design of embedded memories: A survey of technologies, architectures, and optimization techniques

Luca Benini, Alberto Macii, Massimo Poncino

February 2003 **ACM Transactions on Embedded Computing Systems (TECS)**, Volume 2 Issue 1

Publisher: ACM Press

Full text available: pdf(288.44 KB) Additional Information: full citation, abstract, references, citings, index terms

Embedded systems are often designed under stringent energy consumption budgets, to limit heat generation and battery size. Since memory systems consume a significant amount of energy to store and to forward data, it is then imperative to balance power consumption and performance in memory system design. Contemporary system design focuses on the trade-off between performance and energy consumption in processing and storage units, as well as in their interconnections. Although memory design is as ...

Keywords: Embedded systems, embedded memories, integration, memories, nonvolatile, system-on-a-chip, volatile

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2007 ACM, Inc.
Terms of Usage Privacy Policy Code of Ethics Contact Us

Useful downloads: Adobe Acrobat QuickTime Windows Media Player Real Player